

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for tracing an instrumented program using a thread, comprising:
executing a trap instruction to transfer[[ring]] control of the instrumented program to a trap handler, wherein a program counter points to the trap instruction and a next program counter points to a next instruction;
[[to]] obtaining an original instruction associated with a probe;
loading the original instruction into a scratch space;
setting [[a]] the program counter to point to the scratch space;
~~setting a next program counter to point to a next instruction;~~ and
executing the original instruction in the scratch space using the thread, wherein executing the original instruction results in placing the instrumented program in a state equivalent to natively executing the original instruction.
2. (Original) The method of claim 1, further comprising:
determining whether the original instruction is a control-flow instruction; and
emulating a location dependent instruction in a kernel if the original instruction is a control-flow instruction, wherein semantics of the location dependent instruction depend on a location of the original instruction within the instrumented program.
3. (Original) The method of claim 2, further comprising:
updating the program counter and the next program counter using a result from emulating the original instruction in the kernel if the original instruction is control-flow instruction.
4. (Original) The method of claim 1, further comprising:
triggering the probe in the instrumented program.

5. (Original) The method of claim 1, wherein the probe corresponds to a trap instruction.
6. (Original) The method of claim 1, wherein obtaining the original instruction comprises:
searching a look-up table using the program counter, wherein the look-up table contains the original instruction associated with the probe and an address associated with the original instruction.
7. (Original) The method of claim 1, wherein the scratch space is allocated on a per-thread basis.
8. (Original) The method of claim 1, wherein the instrumented program is executed on multi-thread architecture.
9. (Original) The method of claim 1, wherein loading the original instruction comprises using a block copy instruction.
10. (Currently Amended) A computer processor system for tracing an instrumented program, comprising:
 - a program counter configured to store a current address corresponding to a current instruction in the instrumented program;
 - a next program counter configured to store a next address corresponding to a next instruction in the instrumented program;
 - a scratch space arranged to store an original instruction;
 - a thread configured to execute the instrumented program and the original instruction; and
 - a trap handler configured to:
 - halt execution of the thread when a trap instruction is encountered, wherein the program counter points to the trap instruction,
 - to obtain the corresponding original instruction from a look-up table using an address of the trap instruction, and
 - to set the program counter to the scratch space.
11. (Currently Amended) The computer processor system of claim 10, further comprising:
 - a buffer for storing the data.

12. (Currently Amended) The computer processor system of claim 10, further comprising:
a kernel configured to emulate a location dependent instruction if the original instruction is a control-flow instruction, wherein semantics of the location dependent instruction depend on a location of the original instruction within the instrumented program.
13. (Currently Amended) The computer processor system of claim 10, further comprising:
a look-up table configured to store the address and the original instruction.
14. (Currently Amended) The computer processor system of claim 10, wherein the scratch space is allocated on a per-thread basis.
15. (Currently Amended) The computer processor system of claim 10, wherein the instrumented program is executed on multi-thread architecture.
16. (Currently Amended) The computer processor system of claim 10, wherein the trap handler is configured to transfer control to the thread prior to the thread executing the original instruction.